

Penerapan Algoritma *Backtracking* Berbasis *Blind Search* untuk Menentukan Penjadwalan Mengajar

Novriyanto, ST., M.Sc

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Pekanbaru, Indonesia
novri_pci@yahoo.com

M. Zaid.S ,ST

Jurusan Teknik Informatika, Fakultas Sains dan Teknologi
Universitas Islam Negeri Sultan Syarif Kasim Riau
Pekanbaru, Indonesia
mzaid.88@gmail.com

Abstrak—Algoritma pencarian (*searching algorithm*) dilakukan untuk mencari data sesuai argumen kunci yang diterima. Dengan argumen kunci tersebut, hasil pencarian akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*). Penelitian ini menggunakan algoritma *backtracking* untuk menyelesaikan permasalahan dalam menentukan penjadwalan mengajar di Sekolah Dasar (SD) Islam As-Shofa Pekanbaru. Tujuannya adalah untuk membuktikan apakah algoritma tersebut dapat menciptakan jadwal mengajar yang optimal atau tidak. Dari hasil pengujian disimpulkan bahwa algoritma *backtracking* tidak dapat menentukan jadwal mengajar secara keseluruhan. Hal ini disebabkan atas beberapa kendala diantaranya jumlah guru, jumlah mata pelajaran dan jumlah jam yang telah ditentukan.

Kata kunci—Algoritma *backtracking*, *Blind Search*, *DFS*, *Penjadwalan*.

I. PENDAHULUAN

Algoritma pencarian (*searching algorithm*) adalah algoritma yang menerima sebuah argumen kunci dengan langkah-langkah tertentu dalam mencari rekaman dengan kunci tersebut. Setelah proses pencarian dilaksanakan, akan diperoleh salah satu dari dua kemungkinan, yaitu data yang dicari ditemukan (*successful*) atau tidak ditemukan (*unsuccessful*).

Penggunaan algoritma dalam penelitian ini adalah algoritma *backtracking* (runut balik). Algoritma *backtracking* merupakan salah satu metode pemecahan masalah yang berbasis *blind search* (pencarian buta) yang ditelusuri secara *DFS* (*depth – first search*).

Saat ini algoritma *backtracking* banyak diterapkan untuk program *game* seperti permainan teka-teki silang, menemukan jalan keluar dalam sebuah labirin, *math maze*, dan pada kasus lainnya. Pada kasus ini, algoritma *backtracking* akan diterapkan dalam permasalahan penjadwalan mengajar.

Secara umum proses penjadwalan mengajar memiliki kejadian (*event*) dan ruang waktu (*time slot*) yang digunakan untuk menyusun jadwal mengajar. Dengan adanya *event* dan *time slot* algoritma *backtracking* dapat diterapkan untuk

menciptakan sebuah penjadwalan mengajar. Hal ini akan diuji cobakan di SD (Sekolah Dasar) Islam As-Shofa Pekanbaru.

Berdasarkan pembahasan di atas peneliti akan merancang sebuah sistem yang dapat membuktikan apakah algoritma *backtracking* berhasil (*successful*) atau tidak (*unsuccessful*) dalam menciptakan penjadwalan mengajar yang optimal. Untuk itu penulis ingin mengangkat penelitian tugas akhir ini dengan judul “Penerapan Algoritma *Backtracking* Berbasis *Blind Search* untuk Menentukan Penjadwalan Mengajar”.

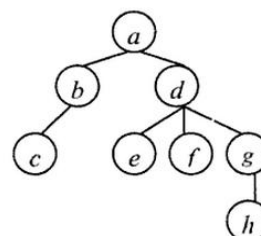
II. LANDASAN TEORI

A. Pohon (*tree*)

Dalam kehidupan sehari-hari kita telah akrab dengan bentuk struktur pohon, khususnya untuk menggambarkan sesuatu yang berjenjang atau berhirarki. Misalnya diagram struktur organisasi (mulai dari pimpinan puncak ke para pegawai di bawahnya), diagram silsilah keluarga, atau jadwal pertandingan dengan sistem gugur.

Sebuah pohon T adalah sebuah himpunan berhingga beranggotakan satu atau lebih data atau *node*, sedemikian hingga dapat ditentukan sebuah *node* khusus, $t \in T$ (disebut *root* untuk pohon T) dan $T - \{t\}$ terpartisi atau terpisah menjadi beberapa sub himpunan T_1, T_2, \dots, T_n , yang masing-masing juga merupakan sebuah pohon yang disebut subpohon atau anak pohon (*subtree*) (Sismoro dkk, 2004).

Dari penjelasan di atas, maka setiap *node* juga merupakan sebuah *root* (*subroot*) untuk suatu subpohon. Untuk lebih jelasnya, perhatikan contoh bentuk pohon berikut:



Gambar 1. Pohon T

Pohon tersebut tersusun atas delapan buah *node*, $T = \{a,b,c,d,e,f,g,h\}$. Pohon T mempunyai *root* a . *Root* a , T terpartisi menjadi dua anak pohon, yaitu $T_1 = \{b,c\}$ dan $T_2 = \{d,e,f,g,h\}$.

Pada pembahasan tentang pohon, sering digunakan istilah-istilah keluarga (*family*). *Node* a dinamakan induk (*parent*) dari *node* b dan d disebut anak (*children*) dari *node* a .

Berkaitan dengan jumlah anak dari suatu *node*, terdapat sebuah definisi pada sebuah pohon. Derajat *node* (*node's degree*) adalah jumlah anak dari suatu *node*. Pada pohon di atas: *node* a berderajat 2, *node* b dan g berderajat 1, *node* d berderajat 3 dan *node* c,e,f serta h berderajat 0. *Node-node* yang berderajat 0 disebut dengan daun (*leaf*) atau *terminated nodes*.

B. Blind Search

Pencarian buta merupakan sekumpulan prosedur yang digunakan dalam melacak ruang keadaan. Pencarian berlangsung sampai solusi terakhir ditemukan. Tujuannya adalah menguji seluruh kemungkinan yang ada untuk menemukan solusi. Digunakan istilah buta karena memang tidak ada informasi awal yang digunakan dalam proses pencarian (Suyanto, 2007).

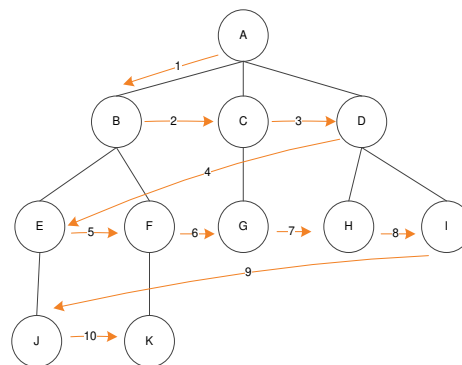
Meskipun teknik pencarian *Blind Search* tidak praktis untuk permasalahan yang besar namun memberikan landasan untuk memahami strategi pencarian informasi (Akerkar, 2005). Semua algoritma pencarian membuat asumsi sebagai berikut:

1. *Node* dapat diperluas oleh beberapa prosedur yang dikenal.
2. Grafik *state-space* adalah pohon. Implikasi adalah bahwa hanya ada satu negara awal atau tunggal dan ada jalan yang unik dari satu *node* ke yang lain.
3. Setiap kali sebuah *node* diperluas untuk menghasilkan penerus atau anak-anak, *pointer* kembali dari anak-anak ke orang tua yang ditetapkan. Jadi ketika sebuah *node* tujuan diperoleh, fitur ini memungkinkan untuk menelusuri jalur solusi.

Pencarian *Blind Search* terdiri atas pencarian melebar pertama (*Breadth First Search*) dan pencarian mendalam pertama (*Depth-First Search*) (Kusumadewi, 2003).

C. Breadth-First Search (BFS)

Pencarian BFS dilakukan pada semua simpul dalam setiap *level* secara berurutan dari kiri ke kanan. Jika pada satu *level* belum ditemukan solusi, maka pencarian dilanjutkan pada *level* berikutnya. Demikian seterusnya sampai ditemukan solusi.



Gambar 2. Pohon pencarian *Breadth-First Search*

1) Skema umum algoritma BFS

Skema umum untuk langkah-langkah pencarian sesuai algoritma *Breadth-First Search* adalah seperti di bawah ini yang dibuat dalam bentuk *pseudo code*.

```

procedure BFS (input v:integer)

Deklarasi
w : Integer
q : antrian

procedure BuatAntrian
(input/output q:antrian)
procedure MasukAntrian
(input/output q:antrian,
input
v: integer)
procedure HapusAntrian
(input/output q:antrian,
input
v: integer)
function AntrianKosong
(input q: antrian)

Algoritma :
BuatAntrian(q)
write(v)
dikunjungi [v] ←true
MasukAntrian(q,v)
while not
AntrianKosong (q) do
    HapusAntrian(q,v)
    fortiap simpul w
yang bertetangga dengan
simpul v
do
    ifnot
dikunjungi[w] then
        write(w)
        MasukAntrian(q,w)
        dikunjungi(w) ←true
    endif
endfor
endwhile
  
```

a) Keuntungan *Breadth-First Search*

- Tidak akan menemui jalan buntu, menjamin ditemukannya solusi (jika solusinya memang ada) dan solusi yang ditemukan pasti yang paling baik.
- Jika ada satu solusi, maka *Breadth-First Search* akan menemukannya, jika ada lebih dari satu solusi, maka solusi minimum akan ditemukan.

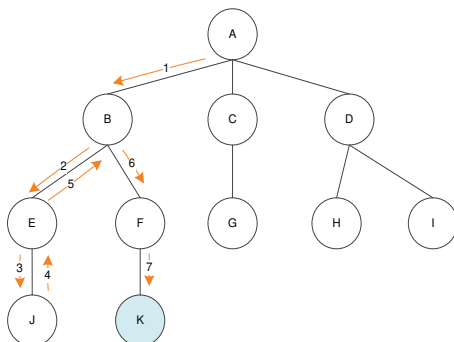
b) Kelemahan *Breadth-First Search*

- Membutuhkan memori yang banyak, karena harus menyimpan semua simpul yang pernah dibangkitkan. Hal ini harus dilakukan agar *Breadth-First Search* dapat melakukan penelusuran simpul-simpul sampai di level bawah.
- Membutuhkan waktu yang cukup lama.

D. Depth-First Search (DFS)

Algoritma DFS pertama kali diperkenalkan oleh Tarjan dan Hopcroft 20 tahun yang lalu. Mereka menunjukkan bagaimana *Depth First Search* (DFS) merupakan metode pencarian secara mendalam dan bagian dari *blind search* atau pencarian buta.

Pencarian dimulai dari level paling pertama, kemudian dilanjutkan ke anak paling kiri pada level berikutnya. Demikian seterusnya sampai tidak terdapat anak lagi atau level yang paling dalam. Jika pencarian belum menemukan solusi, maka dilakukan penelusuran kembali ke *node* sebelumnya dan dilanjutkan ke *node* tetangga. Proses ini diulangi terus hingga menemukan solusi (Kusumadewi, 2003).



Gambar 3. Pohon pencarian *Depth First Search*

1) Skema umum algoritma DFS

Skema umum untuk langkah-langkah pencarian sesuai algoritma *Depth-First Search* adalah seperti di bawah ini yang dibuat dalam bentuk *pseudo code*.

```

procedure DFS (input v:integer)

Deklarasi
w : Integer

Algoritma
write (v)
dikunjungi [v] ← true
for w ← 1 to n do
    if A[v,w]=1 then
        {simpul v dan simpul w
        bertetangga}
        If not dikunjungi [w] then
            DFS (w)
        endif
    endif
endfor

```

a) Keuntungan *Depth-First Search*

- Membutuhkan memori yang relative kecil, karena hanya *node-node* pada lintasan yang aktif saja yang disimpan.
- Secara kebetulan metode *depth-first search* akan menemukan solusi tanpa harus menguji lebih banyak lagi dalam ruang keadaan.

b) Kelemahan *Depth-First Search*

- Memungkinkan tidak ditemukannya tujuan yang diharapkan.
- Hanya akan mendapatkan satu solusi pada setiap pencarian

E. Algoritma Backtracking

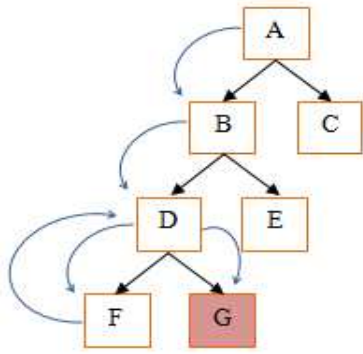
Algoritma *backtracking* pertama kali diperkenalkan oleh D.H Lehmer pada tahun 1950. Dalam perkembangannya beberapa ahli seperti Rwalker, Golomb dan Baumert menyajikan uraian umum tentang *backtrack* dan penerapannya dalam berbagai persoalan dan aplikasi (Teneng dkk, 2010).

Prinsip dasar algoritma *backtracking* adalah mencoba semua kemungkinan solusi yang ada. Perbedaan utamanya adalah pada konsep dasarnya, yaitu pada *backtracking* semua solusi dibuat dalam bentuk pohon solusi (*tree*). Dan kemudian pohon tersebut akan ditelusuri secara DFS (*Depth First Search*) yang merupakan anak dari *Blind Search* sehingga ditemukan solusi terbaik yang diinginkan.

1) Langkah-langkah pencarian algoritma *backtracking*

- Solusi dicari dengan membentuk lintasan dari akar ke daun. Simpul yang sudah dilahirkan dinamakan simpul hidup dan simpul hidup yang diperluas dinamakan simpul-E (*Expand-node*).
- Jika lintasan yang diperoleh dari perluasan simpul-E tidak mengarah kesolusi, maka simpul itu akan menjadi simpul mati dimana simpul itu tidak akan diperluas lagi.
- Jika posisi terakhir ada simpul mati, maka pencarian dilakukan dengan membangkitkan simpul anak yang lainnya dan jika tidak ada simpul *child* (anak) maka dilakukan *backtracking* ke simpul induk.

- Pencarian dihentikan jika kita telah menemukan solusi atau tidak ada simpul hidup yang diperlukan.



Gambar 4. Contoh Algoritma Backtracking

2) Skema umum algoritma Backtracking

Skema umum untuk langkah-langkah pencarian sesuai algoritma Backtracking adalah seperti di bawah ini yang dibuat dalam bentuk *pseudo code*.

```

procedure Backtracking (input
n:integer)

Deklarasi
k : Integer

Algoritma
k ← 1
while k > 0 do
  if (x[k] belum dicoba sedemikian
  sehingga
    x[k] ← T(k) and (B(x[1],
x[2], ..., x[k]) =
    true) then
    If ((x[1], x[2], ... x[k] adalah
    lintasan dari
      Akar ke daun) then
      cetakSolusi(x)
    endif
    k ← k+1 {indeks anggota
    tuple berikutnya}
  else {x[1], x[2], ... x[k] tidak
  mengarah ke
    simpul solusi}
    k ← k-1 {runut balik ke
    anggota tuple
      sebelumnya}
  endif
endif
endwhile

```

F. Algoritma

Kata algoritma diambil dari nama ilmuwan muslim Abu Ja'far Muhammad bin Musa Al-Khwarizmi (780-846) yang banyak menghasilkan karya dalam bidang matematika, disamping karya-karyanya dalam bidang lainnya seperti geografi dan musik (Wahid, 2004).

Algoritma adalah deretan langkah komputasi yang mentransformasikan masukan menjadi keluaran (Munir, 2007). Untuk memulainya perlu adanya perancangan, dan

rancangan tersebut berisi urutan langkah-langkah pencapaian solusi yang biasanya ditulis dalam notasi algoritmik.

G. Penjadwalan

Penjadwalan (*scheduling*) merupakan salah satu kegiatan yang paling penting dalam perusahaan. Penjadwalan adalah pengaturan waktu dari suatu kegiatan operasi, yang mencakup kegiatan mengalokasikan fasilitas, peralatan maupun tenaga kerja, dan menentukan urutan pelaksanaan bagi suatu kegiatan operasi (Herjanto, 2007).

Dalam suatu perusahaan industri, penjadwalan diperlukan antara lain dalam mengalokasikan tenaga operator, mesin dan peralatan produksi, urutan proses, jenis produk, dan pembelian material. Dalam suatu lembaga pendidikan, penjadwalan diperlukan antara lain untuk mengalokasikan ruang kelas, peralatan mengajar, tenaga pengajar, staf administrasi dan pendaftaran penerimaan mahasiswa baru atau ujian. Demikian pula, dalam kegiatan perhotelan, penjadwalan diperlukan dalam pengaturan kamar hotel, ruang seminar atau resepsi, menu makanan, atau acara *entertainment*.

III. ANALISA

Analisa merupakan tahap pemahaman terhadap suatu persoalan sebelum mengambil suatu tindakan atau keputusan. Ini merupakan tahap yang paling penting karena jika terjadi kesalahan pada tahap ini akan menyebabkan kesalahan pada tahap berikutnya. Pada tahapan ini akan dianalisa tentang sistem yang berjalan sekarang dan sistem yang akan dikembangkan.

A. Analisa Sistem Lama

Pada sistem lama, pembuatan penjadwalan mengajar dibuat oleh waka kurikulum yang terdiri atas 3 tugas yaitu:

- Pembuatan pembagian tugas mengajar guru. Tugas ini dibuat dengan cara menentukan mata pelajaran untuk setiap guru yang akan diajarkan pada lokal yang ditentukan oleh waka kurikulum. Data yang diperlukan adalah data guru dan lokal.
- Pembuatan jadwal kegiatan belajar mengajar guru. Tugas ini dibuat sama seperti nomor 1. Perbedaannya adalah dalam tugas ini waka kurikulum menambahkan waktu untuk menentukan guru mengajar di hari apa dan jam berapa. Data yang diperlukan adalah data guru, mata pelajaran, kelas, hari dan jam ajar.
- Pembuatan kontrol mengajar guru. Kontrol mengajar disini adalah menentukan hari dari hari senin hingga jum'at untuk guru piket sesuai ketentuan dari waka kurikulum. Data yang diperlukan adalah data hari dan data guru.

Dari 3 sistem kerja guru ini yang ditugaskan dari pihak sekolah untuk waka kurikulum dalam pembuatan penjadwalan mengajar di SD Islam As-Shofa Pekanbaru.

B. Analisa Sistem Baru

Sistem yang akan dibangun ini adalah Sistem Penjadwalan Mengajar (SPM) dimana sistem penjadwalan ini berisikan tiga tugas waka kurum yaitu jadwal pembagian tugas mengajar guru, jadwal kegiatan belajar mengajar guru dan jadwal kontrol mengajar guru. Namun proses penelusuran algoritma *backtracking* ini diterapkan pada jadwal kegiatan belajar mengajar. Hal ini disebabkan pada jadwal kegiatan mengajar memiliki gabungan dari guru, mata pelajaran, kelas, hari dan jam ajar. Dari adanya gabungan tersebut, algoritma *backtracking* dapat menelusuri kejadian (*event*) ke dalam ruang waktu (*time slot*) yang ada pada jadwal kegiatan belajar mengajar.

1) Analisa Kejadian (*Event*) dan Ruang Waktu (*Time Slot*)

Analisa kejadian (*event*) dan ruang waktu (*time slot*) terdiri dari guru, mata pelajaran, kelas, hari dan jam. Gabungan dari semua ini akan ditampilkan pada jadwal kegiatan belajar mengajar guru.

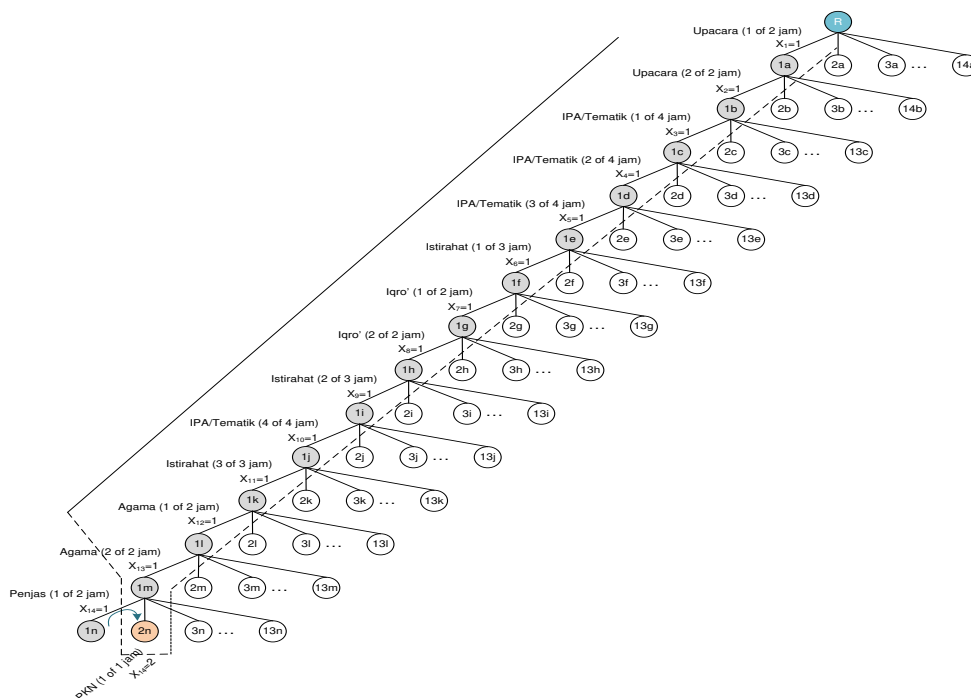
Untuk menerapkan mata pelajaran pada jadwal kegiatan belajar mengajar guru, mata pelajaran akan diberikan batasan-batasan agar jadwal yang akan diciptakan sesuai ketentuan waka kurum.

2) Analisa Penelusuran Algoritma *Backtracking*

Penelusuran algoritma *backtracking* memiliki beberapa tahap. Tahap pertama adalah membuat solusi dalam bentuk pohon (*tree*). Setelah tahap pertama dibuat, maka ditahap kedua adalah penerapan mata pelajaran beserta guru pada kerangka jadwal kegiatan belajar mengajar guru secara DFS.

Penelusuran algoritma *backtracking* dimulai mencari kelas. Kelas di sini dimulai dari kelas IA hingga akhir kelas VIG. Setelah kelas ditemukan, selanjutnya mencari hari, setelah hari ditemukan, selanjutnya mencari jam. Setelah didapat kelas, hari dan jam ditemukan, selanjutnya pemilihan mata pelajaran. Dalam memilih mata pelajaran, mata pelajaran akan diacak untuk mendapatkan salah satu dari 13 mata pelajaran. Setelah mata pelajaran terpilih, mata pelajaran yang terpilih akan diterapkan di jadwal kegiatan belajar mengajar guru. untuk menerapkan mata pelajaran pada jadwal kegiatan belajar mengajar guru juga memiliki batasan-batasan. Setelah semua telah ditelusuri, maka terciptalah jadwal kegiatan mengajar guru.

Dikarenakan penelusuran dimulai dari kelas IA. maka contoh yang akan diterapkan pada analisa ini adalah kelas IA. Berikut gambar penelusuran algoritma *backtracking* untuk kelas IA yang menggunakan kerangka jadwal kegiatan belajar mengajar guru kelas 1 – 2.

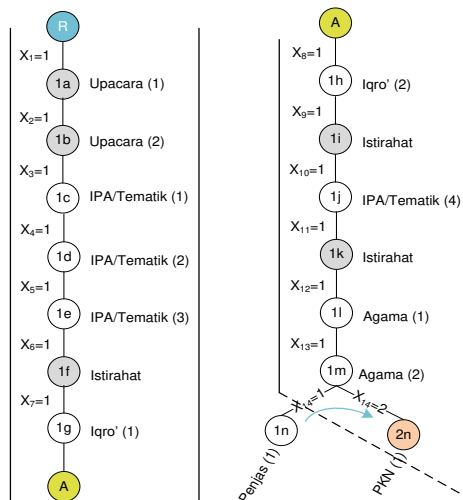


Gambar 5. Pohon Solusi Pencarian Jadwal Kegiatan Belajar Mengajar Hari Senin

Gambar 5 diatas menunjukkan bahwa “R” menandakan bahwa pencarian akan dimulai. Dari level 1 hingga 14 adalah tahapan penyusunan *time slot* di hari senin kelas 1A. Penelusuran memiliki *backtracking* pada level 14. Hal ini

dikarenakan sisa ruang waktu 1 jam diisi mata pelajaran penjas yang memiliki 2 jam dalam sehari sehingga melebihi ruang waktu yang ada di hari Senin, maka dilakukanlah *backtracking*.

Level 14 pada Gambar 5 pada *node* nomor 2 berisikan mata pelajaran PKN (Pendidikan Kewarganegaraan) yang memiliki 1 jam dalam sehari. Pemilihan ini adalah akhir dari pencarian di hari senin sesuai *time slot* yang dimiliki.



Gambar 6. *Node* yang terpilih selama pencarian

IV. SIMPULAN DAN SARAN

A. Simpulan

Setelah melalui tahap analisa dan pengujian pada Sistem Penjadwalan Mengajar ini, maka dapat diambil suatu kesimpulan bahwa dari 30 lokal dan 66 guru serta mata pelajaran yang berjumlah 13, algoritma *backtracking* dapat menentukan penjadwalan mengajar hanya 3 lokal dari 30 lokal. Hal ini disebabkan:

- Keterbatasan jumlah guru dan mata pelajaran yang menjadi salah satu masalah dalam proses penelusuran algoritma *backtracking* yaitu khususnya mata pelajaran PKN kelas 1 dipegang oleh satu guru.
- Sesuai pengujian mata pelajaran PKN ini hanya memiliki kesempatan muncul 3 kali dalam 3 lokal (A, B, C). Sedangkan jumlah kelas 1 adalah 4 kelas (A, B, C, D).
- Dari analisa ini membuktikan bahwa mata pelajaran PKN tidak memiliki kesempatan muncul di kelas 1D. Sehingga mulai dari 1D hingga kelas selanjutnya tidak dapat ditelusuri oleh algoritma *backtracking*.

B. Saran

Agar penelitian ini dapat bermanfaat, khusus untuk penerapan algoritma *backtracking* dalam menentukan penjadwalan ini adalah:

- Diharapkan ada penambahan guru untuk mata pelajaran PKN kelas 1.
- Untuk penelitian selanjutnya algoritma ini kemungkinan besar dapat mengatasi permasalahan penjadwalan mengajar pada studi kasus area universitas khususnya di jurusan.

DAFTAR PUSTAKA

- [1] Akerkar, Rajendra, "*Introduction to artificial Intelligence*", halaman 122, Prentice-Hall, Hindia, 2005.
- [2] Herjanto, Eddy, "*Manajemen operasi*", edisi 3, halaman 307, Grasindo, Jakarta, 2007.
- [3] Kusumadewi, Sri, "*Artificial Intelligence*", halaman 23, Graha Ilmu, Yogyakarta, 2003.
- [4] Munir, Rinaldi, "*Algoritma & Pemrograman dalam Bahasa Pascal dan C*", edisi revisi, halaman 4, Informatika, Bandung, 2007.
- [5] Suyanto, "*Artificial Intelligence*", halaman 15, Informatika, Bandung, 2007.
- [6] Sismoro, Heri, dan Iskandar, Kusri, "*Struktur Data Dan Pemrograman Dengan Pascal*", halaman 124, Andi, Yogyakarta, 2004.
- [7] Teneng, dan Joko, Purwadi, dan Erick, Kurniawan, "Penerapan Algoritma Backtracking pada Permainan Math Maze", *Jurnal Informatika*, Vol. 6 nomor 1, halaman 57, 2010.
- [8] Wahid, Fathul, "*Dasar-dasar Algoritma & Pemrograman*", halaman 1, Andi, Yogyakarta, 2004.